<Design Compiler>

LAB

2012/12/05

1. Copy files

- cp▲-r▲/home/areslab/LAB-DV-12▲./

LAB file list		
Folder	**Name**	**Description**
syn	cpu.v	netlist
MEM	RAM_64B_fast@0C_syn.lib RAM_64B_fast@-40C_syn.lib RAM_64B_typical_syn.lib RAM_64B_slow_syn.lib RAM_64B.v	Memory libraries
tbench	tcpu.v tsmc18.v	testbench verilog model
script	dv_script.tcl .synopsys_dc.setup	script file setup file for Design Compiler
REPORT		Report
WORK		gate level netlist waveform file

2. Synthesis Using <Design Compiler>

- **File preparations:**
  - cpu.v (RTL code)
    - in this step, you must be sure that the RTL code is well verified. For example, c-model and RTL co-verification, coding style checking using n-Lint, code coverage checking using VN,…etc.
  - *.db files (i.e. translated from *.lib on dc_shell platform):
    - RAM_64B_fast@0C_syn.db, RAM_64B_fast@-40C_syn.db, RAM_64B_typical_syn.db, and RAM_slow_syn.db
    - The memory library files (*.lib) are generated by Artisan Memory Compiler
    - The DB files translation
      - > dc_shell-t
      - dc_shell-t> read_lib RAM_64B_slow_syn.lib
      - dc_shell-t>write_lib USERLIB -o RAM_64B_slow_syn.db
  - script file
    - Edit the script file

Step 1: Translate *.lib files to *.db files

Step 2: Run verilog simulation in [tbench]. You can observe the waveforms using nWave before synthesis.

- **ncverilog +access+rwc tcpu.v ../syn/cpu.v**
- **nWave&**
- **open tcpu_rtl.vcd**

Step 3: Edit the files <cpu.v> in [syn] and

<.synopsys_dc.setup> in [script] for synthesis.

- **RTL modification:**
  - All memories and black boxes should be blocked in cpu.v

    //`include "../MEM/RAM_64B.v"
  - Edit .synopsys_dc.setup in [script] folder:
    - Add a "search path" to this file

      set search_path "../MEM/ $search_path"
    - Add the memory *.db files to the "link library" and "target library"

      set link_library "RAM_64B_slow_syn.db ……"

      set target_library "RAM_64B_slow_syn.db…."
- **Edit script file <dv_script.tcl> in [script]:**

Step 1: Specify design

Step 2: Set don't used cells (slow, typical, and fast libraries)

CLKBUF*, CLKINV*, TIE*, SDF*, and SEDF* cells

Step 3: Read files (read cpu.v in verilog format)

"../syn/cpu.v"

Step 4: Set current design as "cpu" and uniquify the design

Step 5: Set operating environment (set both maximum and minimum as "slow" library)

Step 6: Set clk period 10ns

Set don't touch network to "clk" and "rst" signals

Set fix hold to clk

Set clock skews (both hold and setup uncertainties)

Step 7: Set input and output delay 2ns

Set input drive 0.288001pf

Set output load 0.06132pf

Set maximum input transition as 0.11ns

Step 8: Set fanout constraint = 32

Set area constraint = 0

Step 9: Set compile effort as high

- **Open design compiler in [script] and read script:**
  - **>dv**
  - **File -> Execute Script**
  - **dv> exit**
- **After the synthesis, read the area and timing reports in [REPORT]:**

- **Area report in <cpu.area>**
- **Setup time report in <cpu.rpt.maxtiming>**
- **Hold time report in <cpu.rpt.mintiming>**

- **Run verilog simulation after synthesis in [tbench] and observe the waveform using nWave:**
  - **ncverilog +access+rwc**
    tcpu.v ../WORK/cpu.vg ../MEM/RAM_64B.v -v
    tsmc18.v +define+SYNTHESIS_ON
  - **nWave&**
  - **open tcpu_syn.vcd**

實驗報告內容:

1. 實驗目的
2. 附上<cpu.area>與<cpu.rpt.maxitiming>的結果
3. 列出合成後的面積(total cell area) um^2，其中記憶體(RAM_64B)占了多少um^2?
4. 列出合成後的Setup data arrival time與Setup time slack，並解釋data arrival time與slack的意義。